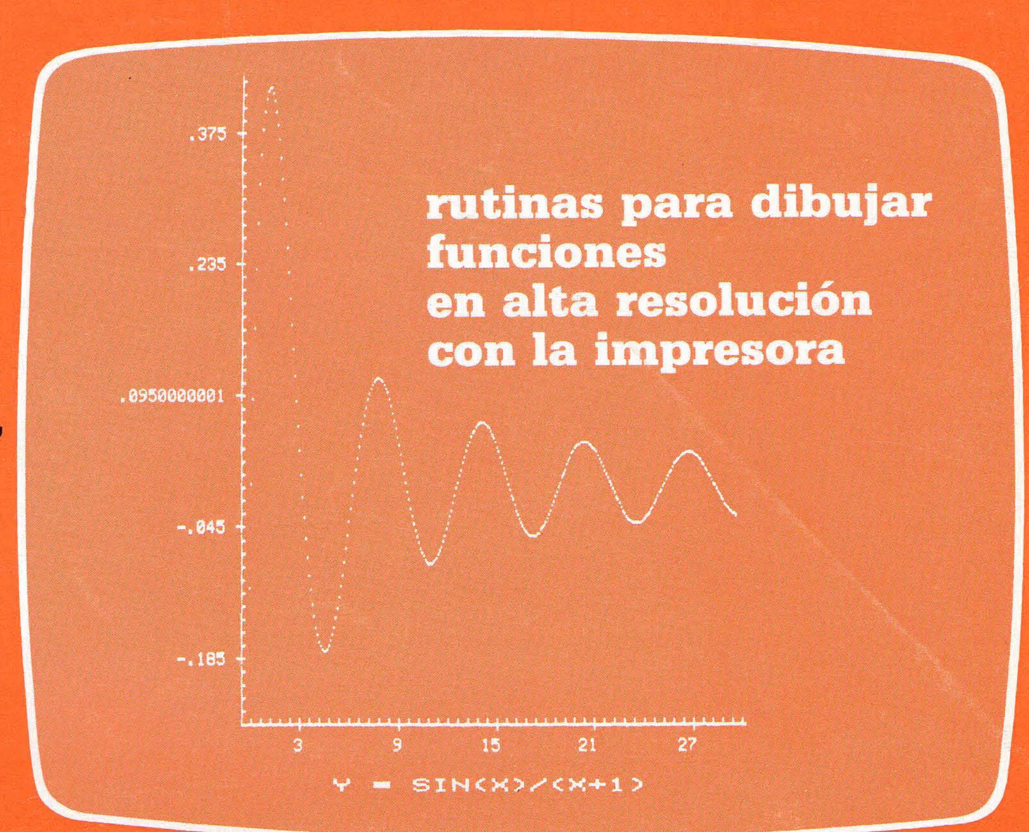
club commodore

Boletín informativo para los usuarios de microordenadores

VIC y CBM



rutinas para dibujar funciones en alta resolución con la impresora

Nº/9

junio 1983

TABLA 1 - CBM - ASCII - CHARACTERS

0	00		64	40	@		128	80		192	C0		
1	01		65	41	a	A	129	81		193	C1	A	
2	02		66	42	b	B	130	82		194	C2	B	
3	03	STOP	67	43	c	C	131	83	[D]LOAD & RUN	195	C3	C	
4	04		68	44	d	D	132	84		196	C4	D	
5	05		69	45	e	E	133	85		197	C5	E	
6	06		70	46	f	F	134	86		198	C6	F	
7	07	BELL*	71	47	g	G	135	87		199	C7	G	
8	08		72	48	h	H	136	88		200	C8	H	
9	09	TAB*	73	49	i	I	137	89	SET TAB*	201	C9	I	
10	0A	LINE FEED	74	4A	j	J	138	8A		202	CA	J	
11	0B		75	4B	k	K	139	8B		203	CB	K	
12	0C		76	4C	l	L	140	8C		204	CC	L	
13	0D	RETURN	77	4D	m	M	141	8D	SHIFT-RETURN	205	CD	M	
14	0E	TEXT*	78	4E	n	N	142	8E	GRAPHIC*	206	CE	N	
15	0F	SET TOP[3]	79	4F	o	O	143	8F	SET BOTTOM[3]	207	CF	O	
16	10		80	50	p	P	144	90		208	D0	P	
17	11	CURSOR DOWN	81	51	q	Q	145	91	CURSOR UP	209	D1	Q	
18	12	REVERSE	82	52	r	R	146	92	REVERSE OFF	210	D2	R	
19	13	HOME CURSOR	83	53	s	S	147	93	CLEAR SCREEN	211	D3	S	
20	14	DELETE CHR.	84	54	t	T	148	94	INSERT CHR.	212	D4	T	
21	15	DELETE LINE*	85	55	u	U	149	95	INSERT LINE*	213	D5	U	
22	16	ERASE END*	86	56	v	V	150	96	ERASE START*	214	D6	V	
23	17		87	57	w	W	151	97		215	D7	W	
24	18		88	58	x	X	152	98		216	D8	X	
25	19	SCROLL UP*	89	59	y	Y	153	99	SCROLL DOWN*	217	D9	Y	
26	1A		90	5A	z	Z	154	9A		218	DA	Z	
27	1B	ESCAPE*	91	5B	[		155	9B		219	DB		
28	1C		92	5C	\		156	9C		220	DC		
29	1D	CURSOR RIGHT	93	5D	]		157	9D	CURSOR LEFT	221	DD		
30	1E		94	5E	↑		158	9E		222	DE		
31	1F		95	5F	←		159	9F		223	DF		
32	20	SPACE	96	60	SPACE[2]		160	A0	SHIFT-SPACE	224	E0	REPEATS[2]	
33	21	!	97	61	!		161	A1		225	E1		
34	22	QUOTE "	98	62	"		162	A2		226	E2		
35	23	#	99	63	#		163	A3		227	E3		
36	24	$	100	64	$		164	A4		228	E4		
37	25	%	101	65	%		165	A5		229	E5		
38	26	&	102	66	&		166	A6		230	E6		
39	27	'	103	67	'		167	A7		231	E7		
40	28	(	104	68	(		168	A8		232	E8		
41	29	)	105	69	)		169	A9		233	E9		
42	2A	*	106	6A	*		170	AA		234	EA		
43	2B	+	107	6B	+		171	AB		235	EB		
44	2C	,	108	6C	,		172	AC		236	EC		
45	2D	-	109	6D	-		173	AD		237	ED		
46	2E	.	110	6E	.		174	AE		238	EE		
47	2F	/	111	6F	/		175	AF		239	EF		
48	30	0	112	70	0		176	B0		240	F0		
49	31	1	113	71	1		177	B1		241	F1		
50	32	2	114	72	2		178	B2		242	F2		
51	33	3	115	73	3		179	B3		243	F3		
52	34	4	116	74	4		180	B4		244	F4		
53	35	5	117	75	5		181	B5		245	F5		
54	36	6	118	76	6		182	B6		246	F6		
55	37	7	119	77	7		183	B7		247	F7		
56	38	8	120	78	8		184	B8		248	F8		
57	39	9	121	79	9		185	B9		249	F9		
58	3A	:	122	7A	:		186	BA		250	FA		
59	3B	;	123	7B	;		187	BB		251	FB		
60	3C	>	124	7C	>		188	BC		252	FC		
61	3D	=	125	7D	=		189	BD		253	FD		
62	3E	<	126	7E	<		190	BE		254	FE		
63	3F	?	127	7F	?		191	BF		255	FF		

EDITORIAL

necesitamos más colaboraciones

Poco a poco van llegando colaboraciones a nuestra Revista, pero todavía son pocas. Debemos poner énfasis en que nuestros lectores no deben dejarse llevar por un mal entendido sentido de vergüenza, pues todos nosotros hemos empezado alguna vez.

Un suscriptor nos pregunta sobre el tipo de programas que nos interesan... Bien, en principio TODOS. No obstante... Juegos tenemos un buen número y creemos que el interés de nuestros lectores se centra más en aquellos programas que representen alguna utilidad. Un inciso: suelen llamarse utilidades en argot informático aquellos programas que facilitan el manejo o la programación del ordenador (ayuda a la programación, rutinas de renumeración, etc...). Este tipo de programas suelen tener una longitud limitada, que, por otro lado, nos interesa porque nuestros listados para que sean claros consumen mucha superficie, tienen un interés general y ayudan a entender el funcionamiento del ordenador. Otro capítulo interesante es el de las aplicaciones, fundamentalmente por dos razones: por su interés en sí mismas y porque pueden ayudar al usuario que empieza a encontrar trabajos para su equipo que no había previsto cuando decidió comprarlo.

Se nos pregunta también por la manera de presentar las colaboraciones. En este aspecto es necesario enviar una cinta (o un disco) con el programa grabado y una explicación lo más exhaustiva posible de su funcionamiento, memoria que necesita, etc... De la preparación del listado para su publicación ya nos encargamos nosotros. La cinta o el disco serán devueltos en el plazo más breve posible con algún otro programa grabado que trate sobre un tema que el mismo autor nos indique en su carta, si lo tenemos en nuestros archivos. Y de momento nada más que volver a nuestra eterna canción: ¡A VER SI COLABORAMOS! pues ésta es la Revista del usuario. ■

TRUCOS

Pokes variados para el VIC-20 y el COMMODORE 64

por **P. MASATS**

Para simular la desconexión del equipo: COMMODORE 64, SYS 64738 y VIC-20, SYS 64802. (¡Estos POKEs son tan variados que hasta incluyen el SYS!). Para impedir el listado de un programa: COMMODORE 64, POKE 775,200 y VIC-20, el mismo. Para retornar a la normalidad: COMMODORE 64, POKE 775,167 y VIC-20, POKE 775,199.

Para inhabilitar la tecla RUN/STOP en el COMMODORE 64: POKE 808,239. En el VIC-20: POKE 788,194. En el Caso del VIC este POKE interfiere el funcionamiento del reloj interno (variables TI y TI$).

Para recuperar el funcionamiento de la tecla RUN/STOP:

En el COMMODORE 64: POKE 808,237.

En el VIC-20: POKE 788,171. ■

VENTANA CBM

modo de utilizar variables del BASIC desde código máquina

por **RAFAEL NAVARRO**

Debido a algunas consultas y opiniones recibidas, considero interesante realizar algunas pequeñas aclaraciones respecto de la representación en estas páginas de los listados de «assembler». El aspecto real de un listado de ensamblador (ayuda simbólica para la escritura de programas en código máquina) tiene, en realidad, un aspecto diferente al de los que se presentan en esta «Ventana». Para facilitar su cotejo, el primer ejemplo de este artículo se presenta bajo formato standard. Como se verá, en el mismo aparecen más datos (número de línea secuencial, dirección de memoria y código hexa —código máquina— de la línea). Si se ha elegido una representación más simbólica es por un motivo fundamental; las áreas disponibles para escritura de programas en c.m. varían sensiblemente de un modelo a otro de COMMODORE, y la intención de esta serie de artículos es la de abarcar, siempre que sea posible, toda la gama de CPUs.

Desgraciadamente, en el número anterior se colaron algunos errores de transcripción en el listado del programa «subrutina de búsqueda de una variable alfanumérica». Por tal causa, repetimos aquí el listado correcto del mismo. FAC y ARG son dos acumuladores empleados por el sistema operativo y el intérprete BASIC para representación de números y aritmética en coma flotante (ver «Ventana» número anterior).

(continúa en la pág. siguiente)

modo de utilizar variables del BASIC desde código máquina

(viene de la pág. anterior)

El tema se complica cuando se trata de localizar variables de tipo «array» o dimensionadas, ya que debe informarse a la rutina BARR de otros datos tales como número de dimensiones y elemento a buscar. Observad el listado adjunto y considerad que el «stack» deberá ser cargado tantas veces como elementos de dimensionado tenga el «array».

La intención de esta sección fija es informaros acerca del mayor número posible de trucos que puedan ser del interés de todos vosotros. Por tanto, nos serán muy útiles todos los comentarios que nos podáis hacer acerca del interés que encontráis en los artículos, así como las sugerencias de nuevos temas a tratar. Hasta el número que viene.

* * *

A la salida de la subrutina los punteros de página cero VARPNT contendrán la dirección de memoria donde se encuentra el elemento.

Para realizar la búsqueda de elementos de «array» alfanumérico bastará introducir modificaciones en la determinación del tipo de variable y realizar un OR del valor del segundo carácter con 128. ■

SUBRUTINA DE BÚSQUEDA DE UN ELEMENTO EN UN «ARRAY» NUMÉRICO

```
BARR                  ; ejemplo de búsqueda de un elemento
                        de ab(0-65535)
      lda # 'a        ; primer carácter nombre variable
      ldy # 'b        ; segundo carácter nombre variable
                      ; ldy # S00 si no hay segundo carácter
      sta varnam      ; almacenamiento del primer carácter
      sty varnam+1
      lda # S00
      sta valtyp      ; tipo de variable numérica
      sta intflg      ; variable coma flotante
      sta dimflg      ; flag de dimensionado
      lda # S01
      sta count       ; número de dimensiones
      ldy # S00-ff    ; peso bajo número elemento
      lda # S00-ff    ; peso alto número elemento
                      ; elemento 0-65535

      pha
      tya
      pha             ; colocar en stack el número de
                        elemento
                      ; esta operación tantas veces como
                        dimensiones
      jmp barr        ; salida a rutina del sistema operativo
```

```
VENTANA8......PAGE 0001

LINE# LOC   CODE        LINE


0001  0000                      *=$033A       ; DIRECCION ORIGEN $033A (DEC. 826) BUFFER 1A CASSETTE DEL CBM8032
0002  033A              VALTYP = $07          ; TIPO DE VARIABLE (0 NUM. 255 ALFA)
0003  033A              INTFL  = $08          ; SI ES NUM., 128 ENTEROS 0COMA FLOTANTE
0004  033A              VARNAM = $42          ; BYTE DONDE EL SISTEMA OPERATIVO REGISTRA EL NOMBRE DE LA VAR.
0005  033A              BVAR   = $C187        ; SUBRUTINA DE BUSQUEDA DE VARIABLES DE SISTEMA OPERATIVO
0006  033A              ; HASTA AQUI, DEFINICION DEL ORIGEN Y ETIQUETAS
0007  033A              BVAL                  ; SUBRUTINA DE BUSQUEDA DE VARIABLES ALFA
0008  033A              ; EJEMPLO DE BUSQUEDA DE AB$
0009  033A  A9 41               LDA #'A       ; PRIMER CARACTER DE LA VARIABLE
0010  033C  A2 42               LDX #'B       ; SEGUNDO CARACTER
0011  033E  85 42               STA VARNAM
0012  0340  8A                  TXA           ; PASAR EL CONTENIDO DEL REGISTRO X AL ACUMULADOR
0013  0341  09 80               ORA #128      ; DISTINTIVO ALFA (SEGUNDO CARACTER = SEGUNDO CARACTER+128)
0014  0343  85 43               STA VARNAM+1
0015  0345  A9 FF               LDA #255
0016  0347  85 07               STA VALTYP    ; VARIABLE TIPO ALFA
0017  0349  4C 87 C1            JMP BVAR
0018  034C                      .END

ERRORS = 0000

SYMBOL TABLE

SYMBOL VALUE
 BVAL     033A    BVAR     C187    INTFL    0008    VALTYP   0007
 VARNAM   0042

END OF ASSEMBLY
```

ficheros relativos en BASIC 4 (I)

por **JORDI SASTRE**

En este artículo no vamos a hablar de ficheros en general (ello es tema para otros artículos) sino que nos limitaremos a los ficheros Relativos.

Un fichero Relativo se compone de registros, cuyo número y longitud han sido definidos por el programador. A cada registro se le accede por su número de orden relativo respecto al principio del fichero (de ahí su nombre).

CREACIÓN DEL FICHERO

Para crear un Relativo, lo primero es decidir cuántos registros desean usarse y qué longitud deben tener. La longitud es la suma de los bytes a almacenar en cada registro más uno, un CHR$(13), que utilizaremos como indicador de «fin de registro». La longitud máxima de los registros es de 254 bytes.

Por ejemplo, vamos a crear un fichero Relativo de 100 registros de 50 bytes cada uno (Cuadro 1).

Mientras se ejecuta la instrucción PRINT #, el LED de error está en rojo (50,RECORD NOT PRESENT) indicando que el registro al que hemos apuntado no existe y, por lo tanto, el DOS debe crearlo junto con todos los anteriores, lo que puede llevar un poco de tiempo.

APERTURA DEL FICHERO

Una vez creado el fichero, ya se le pueden efectuar grabaciones y lecturas de registros pero antes deberemos abrirlo. Se puede leer sólo parte de un registro pero, al grabarse, debe grabarse completo. Para abrir el fichero, utilizaremos la misma instrucción DOPEN # que utilizamos en la creación pero esta vez no es necesario indicar la longitud de los registros (L50), puesto que el DOS ya la sabe. Si se le pone una longitud que no coincide con la definida en la creación de este fichero, el DOS generará un mensaje 50,RECORD NOT PRESENT. Ejemplo de apertura:

DOPEN # 1, «FICHERO»

GRABACIÓN DE REGISTROS

Para grabar registros es necesario seguir al pie de la letra las siguientes normas:

- Antes de cualquier grabación debemos posicionarnos al principio del registro sobre el que se desea grabar (instrucción RECORD #).
- No puede utilizarse más que una instrucción PRINT # para grabar los datos (al utilizar un segundo PRINT #, el DOS grabaría los datos en el registro siguiente). Por tanto, si los datos a grabar no caben en una sola instrucción, antes deberán ponerse en una variable de cadena (por ejemplo B$) para luego grabarla de golpe con un único PRINT #.
- Si los datos a grabar van a leerse posteriormente mediante una instrucción INPUT #, éstos deben separarse entre sí con un CHR$(13). Por ejemplo: 100 PRINT # 1,A1$;CHR$(13);A2$; CHR$(13); ... etc., o bien: 120 B$= A1$+CHR$(13)+A2$+CHR$(13) ... etc./130 PRINT # 1,B$;.
- El último byte grabado en un registro debe ser siempre un CHR$(13) seguido de un punto y coma (;), como puede observarse en los ejemplos.
- Después de grabar el registro mediante el PRINT #, debe posicionarse de nuevo el puntero con una nueva instrucción RECORD #.
- Cuando en un registro se graban menos bytes de los que permite su capacidad, el DOS pone a continuación del último carácter grabado una marca de «Fin de Registro» anulando cualquier contenido posterior que hubiera en el mismo de anteriores grabaciones. Por tanto, no vale grabar sólo parte de un registro, hay que grabarlo entero.

Tomada nota de lo expuesto, véase en el Cuadro 2 un ejemplo de grabación.

LECTURA DE REGISTROS

La lectura de registros ofrece más libertad, puesto que pueden utilizarse tantas GET # s e INPUT # s como se desee; incluso combinaciones de ellas. Para leer un registro sólo hay que posicionarse al principio del mismo y seguidamente leer. Por ejemplo Cuadro 3. ■

CUADRO 1

```
DOPEN # 1,«FICHERO»,L50          : REM 50 = Longitud de los registros
RECORD # 1,100 : PRINT # 1       : REM Creación de 100 registros
DCLOSE # 1                       : REM Fin de Proceso.
```

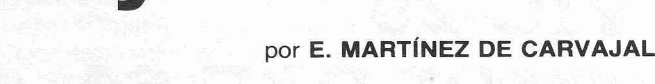
CUADRO 2

```
Y3$=CHR$(13)
DOPEN # 1,«FICHERO»              : REM Apertura del Fichero.
RECORD # 1,(NR)                  : REM Posicionado en registro NR.
B$=A1$+Y3$+A2$+Y3$+
STR$(A3)+Y3$                     : REM A1$ A2$ A3 = Datos a grabar.
PRINT # 1,B$; : RECORD # 1,(NR)  : REM Grabación y reposicionado.
DCLOSE # 1                       : REM Fin de Proceso.
```

CUADRO 3

```
DOPEN # 1,«FICHERO»              : REM Apertura del Fichero.
RECORD # 1,(NR)                  : REM Posicionado en registro NR.
INPUT # 1,A1$,A2$,A3             : REM Lectura de Datos.
DCLOSE # 1                       : REM Fin de Proceso.
```

para las RUN y FILL



por **E. MARTÍNEZ DE CARVAJAL**

do número — 17 — después. Obsérvese que los espacios en blanco que hay entre las letras no se eliminan.

APLICACIÓN PRÁCTICA

Esta función permite determinar la longitud real de una variable alfanumérica. Si tenemos un fichero en el cual todos sus registros están formateados (rellenados) con caracteres en blanco para que tengan la misma longitud, esta función nos da su longitud real antes del formateo. No tiene ninguna utilidad cuando la variable se llena mediante un INPUT por teclado, ya que esta sentencia ya trunca automáticamente el literal que se escriba.

FUNCIÓN FILL$ (Rellenar)

Esta función sirve para llenar un literal con un carácter dado.

ARGUMENTOS DE ENTRADA

X1 Valor ASCII del carácter con que queremos llenar X1$.

X2 Número de caracteres con que queremos llenar X1$. Es la longitud con que quedará.

ARGUMENTOS DE SALIDA

X1$ String rellenado.

EJEMPLO

```
10 X1=48
20 X2=10
30 GOSUB 1000 : FUNCIÓN FILL$
40 PRINT X1$
50 PRINT LEN(X1$)
60 END
RUN
0000000000
10
```

APLICACIONES

Esta función se utiliza normalmente para inicializar registros o variables con un cierto valor. Por ejemplo, variables alfanuméricas que se utilicen para subrayar textos en la pantalla o impresora, para llenar de caracteres en blanco un registro, etc...

EJEMPLO

```
10 X1=ASC('@')
20 X2=22
30 GOSUB 1000 : REM FUNCIÓN FILL$
40 SUB$=X1$ : REM VARIABLE PARA SUBRAYAR TEXTOS EN PANTALLA.
```

```
PROGRAMA:08POS1

1000 REM FUNCION POS
1002 REM
1005 REM DATOS DE ENTRADA
1010 REM X1$=STRING DONDE SE BUSCA
1020 REM X2$=STRING A BUSCAR
1030 REM X1=POSICION INICIAL
1035 REM
1040 REM DATOS DE SALIDA
1045 REM
1050 REM X2=POSICION EN QUE SE ENCU
ENTRA
1100 X2=0
1110 IFX1+(LEN(X2$))-1>LEN(X1$)THEN
RETURN
1120 IFMID$(X1$,X1,LEN(X2$))=X2$THE
NX2=X1:RETURN
1130 X1=X1+1:GOTO1110
READY.
```

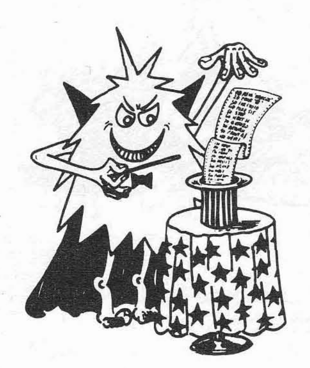
```
PROGRAMA:09TRUN

1000 REM TRUN
1010 REM
1020 REM DATOS DE ENTRADA
1030 REM
1040 REM X1$=STRING A TRUNCAR
1050 REM
1060 REM DATOS DE SALIDA
1070 REM X1$=STRING TRUNCADO
1080 REM
1100 IF ASC(RIGHT$(X1$,1))<>32THENR
ETURN
1110 X1$=LEFT$(X1$,LEN(X1$)-1):GOTO
1100
READY.
```

```
PROGRAMA:09FILL

1000 REM FUNCION FILL$
1010 REM
1020 REM DATOS DE ENTRADA
1030 REM X1=CODIGO ASCII
1040 REM X2=NUMERO DE CARACTERES
1050 REM
1060 REM DATOS DE SALIDA
1070 REM X1$=STRING RELLENADO
1100 X1$=""
1110 FORI=1TOX1:X1$=X1$+CHR$(X2):NE
XTI
1120 RETURN
READY.
```

TRUCOS



trucos y pokes variados para el VIC-20 y el COMMODORE 64

por **P. MASATS**

A continuación damos algunas breves informaciones en torno al VIC y al nuevo COMMODORE 64.

Las dos primeras van para el VIC-20 y son dos rutinas de utilidad que han sido escritas por nuestro colaborador E. Martínez de Carvajal. La primera (en la figura 1 podemos ver el listado y en la 2 un par de ejemplos de su aplicación) permite el volcado del contenido de la pantalla del VIC a la impresora (a esto en el inglés utilizado en informática se le suele llamar un HARDCOPY, o sea una copia sólida). La única limitación que tiene es que sólo permite el volcado de contenidos alfanuméricos o semi-gráficos (los que se obtienen por teclado). En caso de necesitar hacer lo mismo trabajando con el cartucho SUPEREXPANDER o en otro modo en el cual hayamos definido nuestro propio juego de caracteres, podemos utilizar el programa de la figura 3. En éste, las líneas 1-400 constituyen un programa para realizar un dibujo convencional en la pantalla y, a partir de la línea 7000, hasta el final tenemos la subrutina de volcado propiamente dicha. La característica principal de ésta es la de que está «parametrizada», es decir, en las líneas 7010-7180 averigua las posiciones actuales de la memoria de pantalla y del generador de caracteres para conseguir adaptarse a cualquier circunstancia. En la figura 4 se da una muestra de los resultados. En el caso de utilizar la instrucción CIRCLE del SUPEREXPANDER debe realizarse una corrección en los dos radios para compensar el cambio de formato (en algunos casos los círculos se convierten en elipses). Para los usuarios del COMMODORE 64 estamos preparando las correspondientes versiones que publicaremos oportunamente.

```
20 G1=32768
30 IP=7680
40 OPEN1,4
50 FORC=21TO0STEP-1
60 A$=""
70 FORF=0TO22
80 X=PEEK(IP+C+F*22)
90 FORK=0TO7
95 P=PEEK(G1+(X*8)+K):IFP>=128THENP
=P-128
97 P=P+128
100 A$=A$+CHR$(P)
110 NEXTK
120 NEXTF
130 PRINT#1,CHR$(8);A$
140 NEXTC
150 PRINT#1,CHR$(15)
160 CLOSE1
170 STOP
READY.
```

Fig. 1

```
20 G1=32768
30 IP=7680
40 OPEN1,4
50 FORC=21TO0STEP-1
60 A$=""
70 FORF=0TO22
80 X=PEEK(IP+C+F*22)
90 FORK=0TO7
95 P=PEEK(G1+(X*8)+K):
IFP>=128THENP=P-128
97 P=P+128
100 A$=A$+CHR$(P)
110 NEXTK
120 NEXTF
130 PRINT#1,CHR$(8);A$

140 NEXTC
150 PRINT#1,CHR$(15)
160 CLOSE1
170 STOP
READY.
RUN
```

Fig. 2 (A)

```
1234567890+-£QWERTYUIO
P@*↑ASDFGHJKL:;=ZXCVBN
M,./!"#$%&('()
RUN
```

Fig. 2 (B)

* * *

Y ahora unos cuantos POKEs variados:

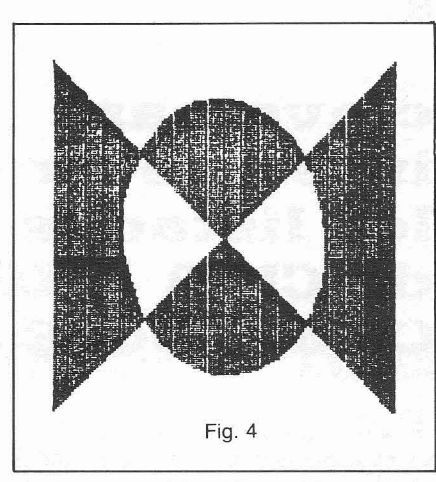
Podemos evitar la acción de la combinación de las teclas RUN/STOP y RESTORE con:

COMMODORE 64: POKE 808,225. — Esto cambia la apariencia de los listados, pero no afecta a la ejecución; también altera el funcionamiento del reloj interno.

Para el VIC-20, junto con el POKE de anulación de la tecla RUN/STOP debe hacerse: POKE 37150,2. Para reactivar estas teclas:
POKE 808,237 — COMMODORE 64 — y POKE 37150,130 en el VIC-20.

Si necesitamos — por alguna razón — prescindir del teclado: COMMODORE 64 y VIC-20: POKE 649,0. Para volver al funcionamiento normal: POKE 649,10 en los dos equipos.

Para terminar con los POKEs, he aquí la manera de cambiar de mayúsculas a minúsculas: En el COMMODORE 64; POKE 53272,peek(53272) OR2. En el VIC-20; POKE 36869,PEEK(36869)OR2. Para recuperar la impresión normal: COMMODORE 64; POKE 53272,PEEK(53272)AND253, y en el VIC-20: POKE36869,PEEK(36869)AND253. ■


Fig. 4

```
1 REM RUTINA PARA EL SUPEREXPANDER
10 GRAPHIC2
20 COLOR1,1,0,0
30 DRAW2,0,0TO1023,1023
40 DRAW2,1023,0TO0,1023
50 CIRCLE2,512,512,300,400
60 PAINT2,0,512
70 PAINT2,1023,512
80 PAINT2,512,400
90 PAINT2,512,800
360 GOSUB7000
370 GRAPHIC0
400 STOP
7000 REM COPIVIDEO
7002 REM ERNESTO MTNZ. DE CARVAJAL
HEDRICH
7010 AV=PEEK(36869):AV=INT(AV/16):A
V=(AV-8)*2
7020 IFPEEK(36866)>=128THENAV=AV+1
7030 AV=AV*512
7050 NL=PEEK(36867):IFNL>=128THENNL
=NL-128
7060 IFNL/2=INT(NL/2)THENCS=8
7070 IFNL/2<>INT(NL/2)THENCS=16
7080 NL=INT(NL/2)
7110 AC=PEEK(36869):X=INT(AC/16):AC
=AC-(X*16)
7120 IFAC>7THENAU=0
7130 IFAC<=7THENAU=1
7140 IFAU=0THENAC=AC-8
7150 IFAU=1THENAC=AC+32
7160 AC=AC*1024
7180 NC=PEEK(36866):IFNC>=128THENNC
=NC-128
8010 OPEN 1,4
8040 FORC=NC-1TO0STEP-1
8050 A$=""
8060 FORF=0TONL-1
8070 X=PEEK(AV+C+F*NC)
8075 FORI=1TOCS/8
8080 FORK=0TO7
8090 P=PEEK(AC+(X*CS)+K+(8*(I-1))):
IFP<=127THEN8100
8095 P=P-128:IFP<=63THENP=P+64
8100 P=P+128
8110 A$=A$+CHR$(P)
8120 NEXTK
8125 NEXTI
8130 NEXTF
8140 PRINT#1,CHR$(8);A$
8150 NEXTC
8160 PRINT#1,CHR$(15)
8170 CLOSE1
8180 RETURN
READY.
```

Fig. 3

BOLETÍN DE SUSCRIPCIÓN - club commodore

NOMBRE .... EDAD ....
DIRECCIÓN ....
POBLACIÓN .... (...........) PROVINCIA ....
TELÉF. .... MARCA Y MODELO DEL ORDENADOR ....
....
APLICACIONES A LAS QUE PIENSA DESTINAR EL EQUIPO ....
....
....

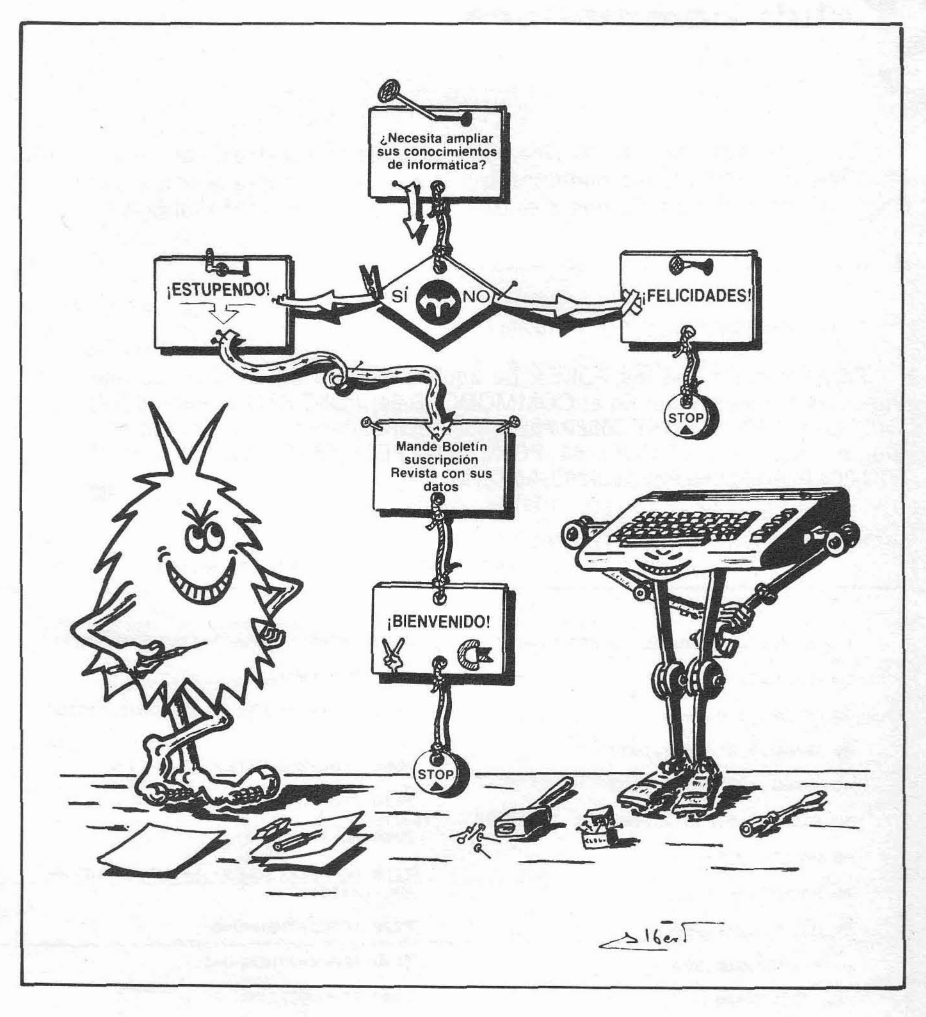
Deseo iniciar la suscripción con el n.° 10

Firma,

(Enviar a la dirección del dorso)

DESEO SUSCRIBIRME A **"CLUB COMMODORE"** POR UN AÑO AL PRECIO DE **1.980** PTAS., QUE PAGARÉ CONTRA REEMBOLSO AL RECIBIR EL NÚMERO CON EL QUE SE INICIA LA SUSCRIPCIÓN. DICHA SUSCRIPCIÓN ME DA DERECHO, NO SÓLO A RECIBIR LA REVISTA (ONCE NÚMEROS ANUALES), SINO A PARTICIPAR EN LAS ACTIVIDADES QUE SE ORGANICEN EN TORNO A ELLA Y QUE PUEDEN SER: COORDINACIÓN DE CURSOS DE BASIC, INTERCAMBIOS DE PROGRAMAS, CONCURSOS, ETC.

clave para interpretar los listados de CLUB COMMODORE



Todos los listados que se publican en esta Revista han sido ejecutados en el modelo correspondiente de la gama de ordenadores COMMODORE. Para facilitar la edición de los mismos en la Revista y para mejorar su legibilidad por parte del usuario, se les ha sometido a ciertas modificaciones mediante un programa escrito especialmente para ello. Para los programas destinados a los ordenadores VIC-20 y COMMODORE 64, en los que se usan frecuentemente las posibilidades gráficas del teclado, se han sustituido los símbolos gráficos que aparecen normalmente en los listados por una serie de letras entre corchetes [] que indican la secuencia de teclas que se deben pulsar para obtener el carácter deseado. A continuación se da una tabla para aclarar la interpretación de las indicaciones entre corchetes:

[CRSRD] = Tecla cursor hacia abajo (sin SHIFT)
[CRSRU] = Tecla cursor hacia arriba (con SHIFT)
[CRSRR] = Tecla cursor a la derecha (sin SHIFT)
[CRSRL] = Tecla cursor a la izquierda (con SHIFT)
[HOME] = Tecla CLR/HOME (sin SHIFT)
[CLR] = Tecla CLR/HOME (con SHIFT)

Las indicaciones [BLK] a [YEL] corresponden a la pulsación de las teclas de 1 a 8 junto a la tecla CTRL. Lo mismo sucede con [RVSON] y [RVSOF] respecto a la tecla CTRL y las teclas 9 y 0.

El resto de las indicaciones constan de la parte COMM o SHIF seguidas de una letra, número o símbolo — por ejemplo [COMM+] o [SHIFA] —. Esto indica que para obtener el gráfico necesario en el programa deben pulsarse simultáneamente las teclas COMMODORE (la que lleva el logotipo) o una de SHIFT y la tecla indicada por la letra, el número o el símbolo, en el ejemplo anterior: COMMODORE y + o SHIFT y A, respectivamente. ■



PROGRAMAS EN DISCO

D-1001 Agenda 5.000
Permite contener 4 páginas de información de 114 personas, entidades, etc. Pudiendo añadir, cambiar o borrar la información que se desee.
En castellano, necesita ampliación de 8K. Presentado en caja.

D-1002 QSL 3.000
Registro de contactos para radioaficionados, versión en disco.
Necesita ampliación de 8K. Presentado en caja.

D-1003 Test Demo 3.000
Programa de test para la unidad de disco. Nota: este disco es el que se suministra con la unidad de disco VIC-1540.

D-1004 Assembler 5.000
Editor y compilador en BASIC para rutinas en lenguaje máquina del 6502. Necesita ampliación de 3K. Acompañado de manual en inglés.

D-1005 English language 2.500
Versión en disco del programa en cinta, dado que al manejar ficheros resulta mucho más rápida su utilización.
Necesita ampliación de 8K. Presentado en caja.

D-1006 Quiz Master 2.500
Versión en disco del programa en cinta. Necesita ampliación de 8K. Presentado en caja.

D-1007 Matemáticas 1 (nivel BUP) 2.500
Versión en disco del programa en cinta. Necesita ampliación de 8K. Presentado en caja.

3301 Simplicalc 13.000
Hoja de Trabajo Electrónica 60 columnas por 100 filas Max. Necesita ampliación de 16K.

3304 Vic File 13.000
Bases de datos para el VIC-20 con posibilidad de definir el formato de los campos. Necesita ampliación de 16K. Ideal para fichero de clientes.

3305 Vic Writer 13.000
Proceso de texto muy potente y rápido. Necesita ampliación de 8K ó 16 K.

PROGRAMAS EN CINTA

Programas Técnicos

C-128 Programación lineal 1.000
Método simplex. Cálculo del valor de las variables que satisfaciendo las restricciones hacen máxima o mínima una función. En castellano. Presentado en caja.

C-129 Matrices 1.000
Suma, resta, multiplicación, multiplicación por un escalar e inversión de matrices. En castellano. Presentado en caja.

C-131 Regresiones I 1.000
Contenido: Regresión lineal: Cálculo por mínimos cuadrados de la recta que se ajusta mejor a una nube de puntos.
Regresión múltiple: Variable dependiente en función de N variables independientes de grado 1. En castellano. Presentado en caja.

C-132 Regresiones II 1.000
Contenido: Regresión de orden N: Variable dependiente en función de variable independiente de grado N.
Regresión exponencial: Ajuste nube de puntos a una curva exponencial.
Regresión geométrica: Ajuste de una nube de puntos a una curva geométrica. En castellano. Presentado en caja.

C-133 Estadística I 1.000
Distribución normal, Poisson, binomial, chi-cuadrado, Student, F de Snedecor. En castellano. Presentado en caja.

C-134 Estadística II 1.000
Contenido: Cálculo de la media, varianza y desviación tipo, tanto de la muestra como de la población, estando los datos agrupados o no. Test de chi-cuadrado y test de Student. En castellano. Presentado en caja.

C-135 Sistemas 1.000
Contenido: Resolución de sistemas de N ecuaciones con N incógnitas.
Resolución de ecuaciones de grado 2 dando las soluciones tanto reales como complejas.
Cálculo de permutaciones y de combinaciones. En castellano. Presentado en caja.

C-137 Integración 1.000
Cálculo de integrales por el método de Gauss.
Cálculo de derivadas de una función en un punto. Interpolaciones tanto lineales como curvilíneas. En castellano. Presentado en caja.

Programas Educativos

C-125 Hangmath 1.500
Se trata de averiguar la multiplicación entre 2 números (tanto los números como la multiplicación y el resultado final), procurando no ser ahorcado. Las instrucciones son en castellano.

C-140 Skymath 1.500
Sumas y restas para chicos muy jóvenes saliendo naves espaciales para motivarlos. Una manera de aprender jugando. 3K.

C-141 Space Division 1.500
20 preguntas sobre divisiones muy sencillas. En función de los resultados correctos, despega un cohete que llegará más o menos lejos. 3K.

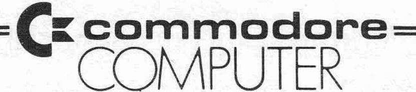
C-143 English Language 2.000
Especialmente adecuado para las personas que, poseedoras de un buen nivel de inglés, quieren alcanzar cotas superiores. Necesita ampliación de 8K. Presentado en caja y acompañado de su correspondiente manual.

C-145 Mastermind 3.000
Participe en el famoso concurso de la BBC, de preguntas y respuestas sobre temas variados.
Este cassette va acompañado del sistema operativo y de un file de demostración. Necesita ampliación de 8K. Presentado en caja.

C-146 Matemáticas I (nivel BUP) 2.000
Contiene sistema operativo y 7 cuestionarios sobre diferentes temas abarcando geometría, aritmética, matemáticas generales, álgebra, etc. Necesita ampliación de 8K ó 16K. Presentado en caja.

PROGRAMAS COMERCIALIZADOS POR "MICROELECTRÓNICA Y CONTROL, S.A."
Puede encontrarlos en su distribuidor más próximo

microelectrónica y control s a

El mejor ordenador personal del Mundo

El COMMODORE 64

Este es el nuevo ordenador personal COMMODORE 64. Un gigante de 40 cm, con un precio casi tan pequeño como su tamaño.

Nadie hasta ahora había logrado ofrecerle 64 K de memoria, 40 columnas en pantalla, 8 sprites y un sonido de auténtica maravilla por sólo 110.000,— ptas. Claro que tampoco todo el mundo es el líder mundial en microordenadores.

COMMODORE sabe perfectamente que para seguir siendo el número uno, tiene que estar constantemente en vanguardia.
De calidad. De precios. De todo. Para ello investigamos constantemente.

Afortunadamente nuestra labor se ve plenamente recompensada cuando vemos, como lo demuestra el cuadro comparativo, que nuestro más directo competidor cuesta nada menos que un 100% más caro. Y ello sin reunir todos los adelantos técnicos del COMMODORE 64.

---

1. Capacidad total de memoria RAM de 64 K. Interpretador BASIC extendido y sistema operativo residentes en ROM.

2. Dotado del más potente chip sintetizador de sonido diseñado hasta hoy, el COMMODORE 64 ofrece 3 voces totalmente independientes con una gama de 9 octavas. El programa puede controlar la envolvente, la afinación y la forma de onda de cada voz,

convirtiendo al COMMODORE 64 en el mejor simulador de instrumentos.

3. Conectable directamente a toda una gama de periféricos, incluyendo unidad de discos, impresora de matriz de puntos o de margarita, plotter, comunicaciones locales y remotas..., y mucho más.

4. Pantalla de alta resolución en color con 320 × 200 puntos directamente direccionables. Capacidad en modo carácter de 25 líneas por 40 columnas.

5. El chip de video, único en su género, permite el uso de 8 «Sprites» (figuras móviles en alta resolución y color). Los «Sprites» pueden moverse independientemente por programa de «pixel» en «pixel».

6. A cada «Sprite» se le asigna por programa un nivel de prioridad en caso de cruce con otro, consiguiendo efectos tridimensionales, existiendo también detección automática de colisiones.

7. Teclado profesional con mayúsculas y minúsculas, más 62 caracteres gráficos, todos ellos disponibles en el teclado y visualizables en 16 colores, en forma normal o bien en video invertido.

8. Encontrará a su disposición una completa gama de programas profesionales, incluyendo proceso de textos, sistemas de información, modelos financieros, contabilidad y muchas más aplicaciones.

9. Están en fase de desarrollo asimismo otros lenguajes tales como LOGO, UCSD PASCAL, COMAL, ASSEMBLER, etc. Todos los programas existentes de la gama COMMODORE, desde el VIC-20 hasta los modelos CBM pueden ser adaptados fácilmente.

10. Posibilidad de inserción de cartuchos con programas grabados en ROM, tanto profesionales como para educación y ocio.

11. Opción de un segundo procesador Z-80 para trabajar con sistema operativo CP/M (R).

EL COMMODORE 64 Y SU MAS DIRECTO COMPETIDOR

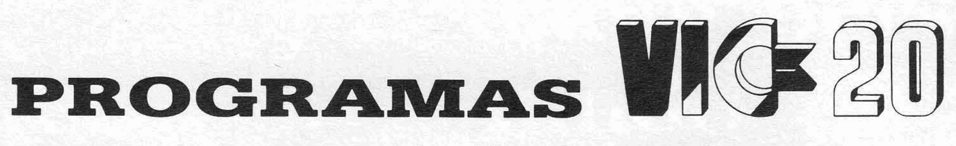
OPCION DE BASE	COMMODORE 64	Más directo competidor
Precio	110.000,— ptas.	El doble
Memoria usuario	64 K	48 K
Teclado profesional	SI	SI
Teclado con caracteres gráficos	SI	NO
Minúsculas	SI	NO
Teclas de función	SI	NO
Máxima capacidad disco	170 K a 1 M	143 K
AUDIO		
Generador de sonido	SI	SI
Sintetizador de música	SI	NO
Salida HI-FI	SI	NO
VIDEO		
Salida monitor	SI	SI
Salida para TV	SI	EXTRA
PERIFERICOS		
Cassette	SI	SI
Periféricos inteligentes	SI	SI
Bus serie	SI	NO
SOFTWARE		
Opción CP/M (R)	SI	SI
Ranura cartucho externo	SI	NO

commodore
COMPUTER

PARA MAS INFORMACION
DEL COMMODORE 64,
LLAMAR O ESCRIBIR A:
MICROELECTRONICA Y CONTROL
c/ Taquígrafo Serra, 7, 5.º. Barcelona-29
Tel. (93) 250 51 03
c/ Princesa, 47, 3.º, G. Madrid-8
Tel. (91) 248 95 70

Nombre ....
Dirección ....
Tel. ....
Población ....

SLOGAN

PROGRAMAS 

Programas Aplicaciones

C-130 **Caja** 2.000
Dentro de una cuenta bancaria permite abrir hasta 100 subcuentas con su correspondiente debe y haber. Preparado para funcionar si se desea con impresora, unidad de discos y evidentemente cassette. Necesita ampliación 16K. Presentado en caja.

C-136 **Dieta** 2.000
En función de la edad, sexo, altura, configuración y actividad desempeñada por la persona, da el peso ideal. Pregunta si se desea seguir un régimen, y en función de la actividad física desarrollada da el peso que se debe tener y la pérdida diaria que se debe alcanzar hasta llegar a dicho peso óptimo. Está en castellano y necesita ampliación de 8K. Presentado en caja.

C-139 **Vicalc** 1.500
El VIC-20 se comporta como una calculadora y utiliza la notación polaca inversa. Manual de instrucciones en castellano.

C-142A **Interface de RTTY y CW y cassette de CW** . 25.000
Especial para radioaficionados.

C-142B **Cassette de RTTY** 2.500
Para utilizarlo es necesario el interface de RTTY y CW. Especial radioaficionados.

C-144 **Quiz Master** 2.000
Programa para preparación de cuestionarios. Especialmente adecuado para escuelas. En castellano. Necesita ampliación de 8K.

Programas de Juegos

C-201 **Codemaker** 1.500
El VIC-20 inventa un código secreto, teniendo que descubrirlo. Instrucciones en castellano y el programa en inglés. No necesita ampliación.

C-202 **Wall Street** 1.500
Partiendo de una situación financiera inicial, puede comprar y vender acciones diariamente en función de los precios que van variando. Instrucciones en castellano. Programa en inglés. No necesita ampliación.

C-203 **Simple Simon** 1.500
El VIC-20 va generando una serie de notas musicales que el usuario debe repetir en la misma secuencia. El grado de dificultad va aumentando y cuando se comete un error se vuelve a iniciar el juego. Instrucciones en castellano. Programas en inglés. No necesita ampliación.

C-2041 **Damas** 1.500
Juego de Damas especialmente adecuado para niños entre los 7 y los 10 años. En inglés. No necesita ampliación.

C-205 **Alien Blitz** 1.500
Fabuloso juego de invasores para el VIC-20. No necesita ampliación.

C-206 **Kosmic Kamikaze** 1.500
Ataque masivo de naves extraterrestres, pudiéndole salvar tan sólo sus habilidades y reflejos. Necesita ampliación de 3K u 8K, dado que hay dos versiones del juego.

C-207 **Star Wars** 1.500
Tan solo dispone de 1 minuto para destruir al enemigo!!, siendo difícil el tener colocado los invasores en el centro de la pantalla para abatirlos. No necesita ampliación.

C-208 **Amok** 1.500
Solamente con su habilidad y reflejos, el hombre dirigido por usted podrá esquivar el ataque del enemigo y aniquilarlo. No necesita ampliación.

C-202 **The Alien** 1.500
¡El comecocos en acción! Necesita una ampliación de 3K.

C-210 **Invader Fall** 1.500
Defenderá su posición frente al avance progresivo de naves espaciales. Debe conseguir la destrucción total antes de la llegada a sus líneas. No necesita ampliación.

C-211 **A-MAZ-ING** 1.500
Piérdase por un laberinto poblado de fantasmas. Necesita ampliación de 3K.

C-212 **Math-Hurdler, Monster Maze** 1.500
Math-Hurdler: Tiene que resolver la operación matemática por usted escogida con el grado de dificultad y rapidez que desee. Monster Maze: Debe llegar al otro extremo del laberinto evitando que sus enemigos se lo coman. No necesita ampliación.

C-213 **Golf** 1.500
Juegue al golf sentado cómodamente en su casa frente a la pantalla del televisor. Necesita 3K.

C-215 **VIC Games II** 1.500
Fire: Apague un incendio con un helicóptero.
Draw: Dibuje en la pantalla con alta resolución.
Race: Intente imitar a Fangio en espectacular carrera de coches. No necesita ampliación.

BIBLIOGRAFÍA

Manual usuario VIC-20 800
Nociones de funcionamiento del VIC-20 y del lenguaje BASIC.

Curso Introducción BASIC: Parte I 2.500
De una manera clara y sencilla, partiendo de cero, se va enseñando el lenguaje del futuro. Este libro va acompañado de dos cassettes con 17 programas en castellano.

Curso Introducción BASIC: Parte II 2.500
Profundización de lo tratado en Parte I, especialmente en creación de ficheros, manejo de subrutinas, efectos musicales, aspectos animados, etc.

VIC Revealed 1.500
Libro que explica con detalle el funcionamiento interno del VIC-20. Incluye esquemas. Libro en inglés.

Programers Reference Guide 2.000
Profundización en el manejo del VIC-20. Incluye referencias a rutinas lenguaje máquina. Libro fotocopiado en inglés.

PROGRAMAS COMERCIALIZADOS POR "MICROELECTRÓNICA Y CONTROL, S.A."
Puede encontrarlos en su distribuidor más próximo

microelectrónica y control s.a 



rutinas para dibujar funciones en alta resolución con la impresora

por **P. MASATS**

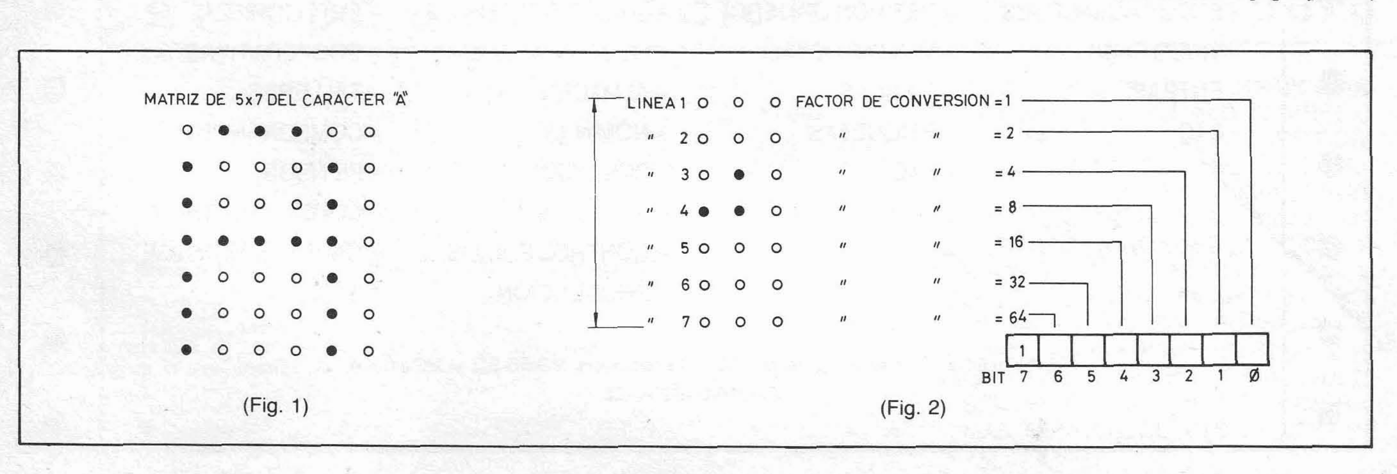
En este artículo presentamos una rutina que permite dibujar en la impresora del VIC-20 una función matemática estableciendo una serie de parámetros (fórmula de la función, valores iniciales e incrementos de X e Y) y haciendo un GOSUB 1000 la impresora nos realizará un dibujo de la función con sus ejes cifrados y un título opcional.

Las relaciones matemáticas se pueden entender y analizar mejor si se dibujan y un ORDENADOR de bajo costo como el VIC-20 puede manejar funciones de una manera potente y flexible. El único aspecto que queda por resolver es el de conseguir un sistema para que estas funciones «dejen rastro» de una manera permanente. La impresora VIC-1515, además de poder escribir con letras y números de manera convencional, permite realizar dibujos de una manera similar a la que utiliza el VIC en la pantalla, es decir, de alguna manera, se pueden definir los caracteres que se van a imprimir. Veamos rápidamente cómo trabaja la impresora de manera general y específicamente para dibujar.

La impresora VIC-1515 funciona en tres modos diferentes: cuando se pone en marcha la unidad o se le manda un CHR$(15) ésta se halla en el modo de caracteres standard que permite la impresión de los símbolos alfanuméricos y gráficos de uso habitual en la pantalla. Este modo de funcionamiento es el utilizado para obtener listados y datos alfanuméricos. En caso de necesitar la impresión de los caracteres de la parte derecha de las teclas, se pueden obtener mediante una dirección secundaria especial en el comando OPEN. Se pueden conseguir caracteres de doble ancho (en este caso sólo caben 40 caracteres en una línea) mandando a la impresora un CHR$(14). Esto expande cada carácter horizontalmente mejorando la legibilidad. Se puede utilizar con todos los caracteres alfanuméricos. El modo gráfico se obtiene enviando un CHR$(8). De esta manera se obtiene el control de cada punto individual en la línea de impresión. Cuando la impresora — en modo alfanumérico — recibe el código de un carácter en particular, primero consulta una memoria ROM que lleva incorporada (el concepto es muy similar al del generador de caracteres en el funcionamiento del video) para ver cuál es la configuración del carácter que debe escribir. La ROM suministra información para una matriz de 5 × 7 puntos (ver la figura 1) y la lógica de la impresora deja en blanco la sexta columna para separar el carácter actual del siguiente. Al enviar un código de 8 bits se puede escribir cualquiera de los caracteres programados en la ROM.

En el caso del modo gráfico, se nos permite «saltarnos» la consulta a la ROM y enviar a la cabeza de impresión la columna que deseamos dibujar (ver figura 2). De esta manera el código no se interpreta como un carácter

(continúa en la pág. siguiente)



(Fig. 1)

(Fig. 2)

a imprimir sino que es procesado como un carácter de control si el bit 7 es 0 (si el valor decimal es menor que 128), o se envía directamente al cabezal de impresión si el bit 7 es igual a 1 (o sea, si el valor está entre 128 y 255). En este último caso los bits del 0 al 6 determinan directamente en qué línea se va a poner un punto o no. Por ejem-

```
READY.

1000 B$=CHR$(16)+CHR$(50)+CHR$(48):
C$=CHR$(13)

1010 D1$=CHR$(171)+CHR$(8):D2$=CHR$
(123)+CHR$(8):SP=128

1020 OPEN1,4:PRINT#1:PRINT#1

1030 YR=YI*7:FORRN=0TO49:RV=YS+(50-
RN)*YR

1040 FORCN=0TO300:A%(CN)=SP:NEXTCN

1050 IFRN-10*INT(RN/10)<>5THENPRINT
#1,C$;B$;D1$;GOTO1090

1060 CN=RV-3*YI:IFABS(CN)<0.5*YITHE
NCN=0

1070 PRINT#1,C$;:E$=STR$(CN):FORN=1
TO(19-LEN(E$)):PRINT#1," ";:NEXTN

1080 PRINT#1,E$;B$;D2$;:CE=0

1090 FORCN=0TO299

1100 RS=RV-FNY(XS+CN*XI):IFRS<0THEN
1130

1110 DR=INT(RS/YI):IFDR>6THEN1130

1120 A%(CN)=128+2↑DR:CE=CN

1130 NEXTCN

1150 CE=CE+1:A%(CE)=13

1160 FORCN=0TOCE:PRINT#1,CHR$(A%(CN
));:NEXTCN

1170 NEXTRN:PRINT#1,C$;B$;CHR$(173)
;

1180 FORCN=0TO50:IFCN-10*INT(CN/10)
<>5THENPRINT#1,CHR$(177);:GOTO1200

1190 PRINT#1,CHR$(123)

1200 NEXTCN:PRINT#1

1210 FORNN=5TO45STEP10:F$=STR$(NN+1
5):PRINT#1,CHR$(16);MID$(F$,2,2);

1220 XV=XS+(NN+0.5)*XI*6:IFABS(XV)<
0.5*XITHENXV=0

1230 E$=STR$(XV):L=LEN(E$):IFL>10TH
ENL=10

1240 FORN=0TOINT(5-L/2):PRINT#1," "
;:NEXTN

1250 PRINT#1,E$;:NEXTNN:PRINT#1:PRI
NT#1

1260 PRINT#1,CHR$(14);:L=LEN(T$):IF
L>40THENL=40

1270 FORN=1TOINT(23-L/2):PRINT#1,"
";:NEXTN

1280 PRINT#1,T$;CHR$(15):PRINT#1:PR
INT#1:CLOSE1

1290 RETURN

READY.
```

(Fig. 3)

plo, para dibujar un punto en la línea 4 (ver figura 2) el valor a enviar debe ser 128 + 8, es decir CHR$(136). Para dibujar a continuación en las líneas 3 y 4 se debe mandar 128 + 8 + 4, o sea CHR$(140). Como sea que se van enviando caracteres sucesivos para cada columna independientemente se puede simular el funcionamiento en modo alfanumérico mandando sucesivamente los CHR$ de 254, 137, 137, 137, 254 y 128 para dibujar la «A» de la figura 1.

La impresora además controla automáticamente el buffer de transferencia de datos en modo gráfico, puede tomar hasta 481 bytes de datos para dibujar una línea (es decir, el CHR$(8) para entrar en el modo gráfico y 80 (caracteres) * 6 (columnas por carácter) = 480 columnas), pero el buffer interno de la impresora tiene una capacidad de 90 bytes por lo que dibujará la línea «a trozos». Sin embargo, esto no tiene ninguna consecuencia

```
100 DIMA%(300)

110 XS=-0.3:YS=-0.249:YI=0.002:XI=0
.1

120 DEF FNY(X)=SIN(X)/(X+1)

130 T$="Y = SIN(X)/(X+1)"

140 GOSUB1000:END

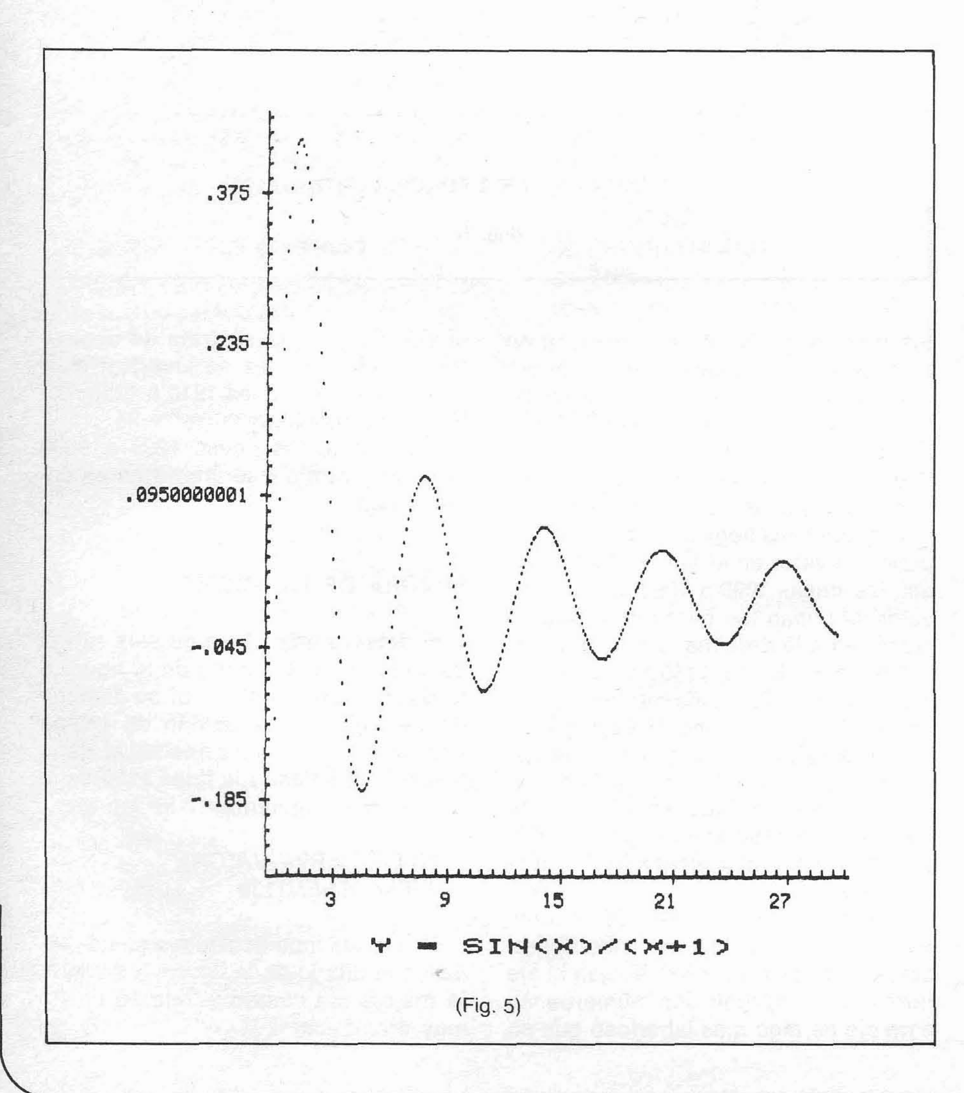
READY.
```

(Fig. 4)

para la programación exterior. Para evitar el espacio que normalmente se deja entre líneas, el avance de papel es de dos tercios de el alfanumérico, con lo que las líneas sucesivas quedan impresas sin solución de continuidad.

La subrutina de la que se da el listado en la figura 3 ha sido escrita para utilizar este concepto de gráfico. Con

(continúa en la pág. siguiente)



(Fig. 5)

rutina para dibujar funciones en alta resolución con la impresora

(conclusión)

su uso y el del programa de la figura 4 se ha obtenido el dibujo de la figura 5. En el listado de la figura 4 la línea 110 está encargada de situar los parámetros más importantes. XS e YS son los valores iniciales para la función. XI e YI son los valores en los que se incrementará la función. En la línea 120 escribimos la función que queremos representar utilizando DEF FN. A continuación en la línea 130 la variable T$ almacena la cadena alfanumérica que será el título que se imprime al pie del dibujo. La rutina utiliza alrededor de 1600 bytes con lo que queda espacio incluso en un VIC sin expansión.

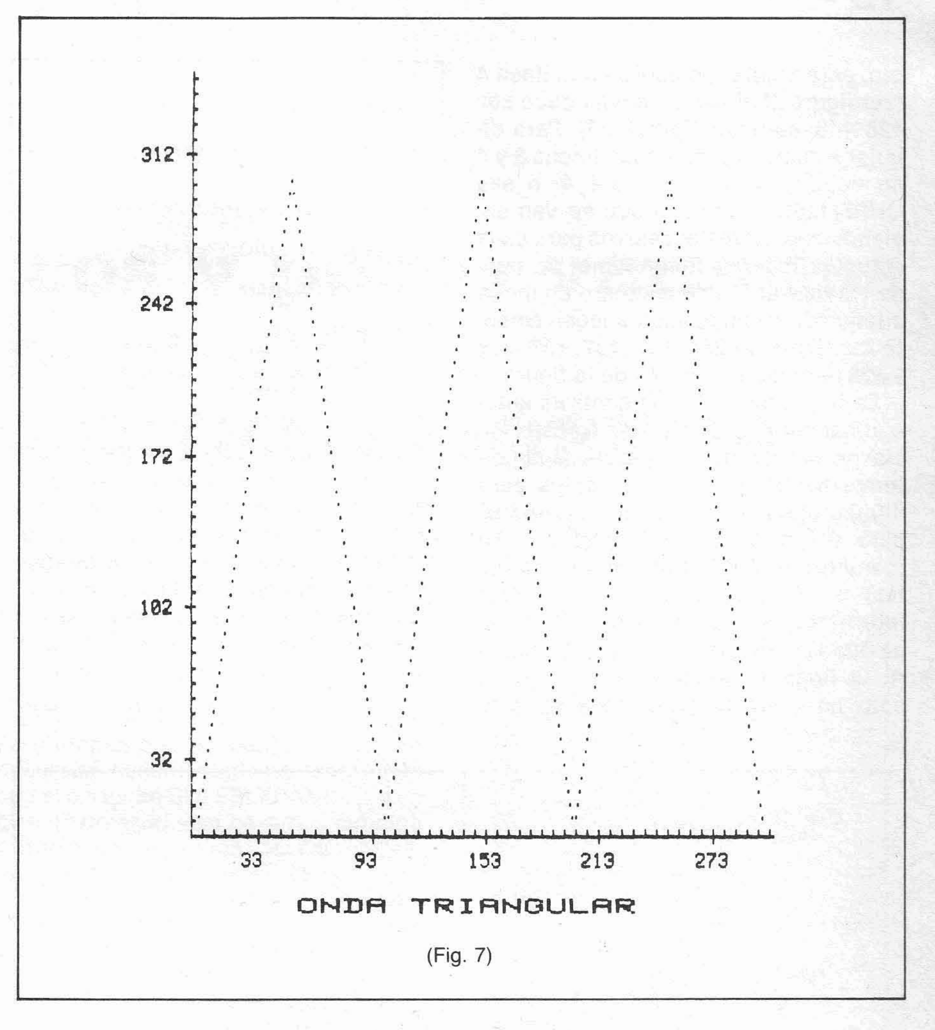
EL PROGRAMA

Las líneas 1000 a 1020 realizan la inicialización general. B$ es el equivalente en la impresora de la función TAB y hace avanzar a la cabeza de impresión a la posición 20 para colocar el eje vertical. C$ coloca la impresora en modo carácter standard. D1$ y D2$ imprimen los caracteres que forman el

```
READY.

100 DIMA%(300),VA(300)
110 XS=0:YS=0:YI=1:XI=1
120 VA(0)=0:FORCN=0TO299:IFCN-100*I
NT(CN/100)>=50THENIN=-6:GOTO140
130 IN=6
140 VA(CN+1)=VA(CN)+IN:NEXTCN
150 T$="ONDA TRIANGULAR"
160 GOSUB1000:END
READY.
```

(Fig. 6)



(Fig. 7)

eje vertical y sitúan la impresora en modo gráfico, preparada para dibujar.

La línea 1030 empieza el bucle externo. YR es el incremento en el eje Y por línea de impresión (7 líneas). La línea 1040 inicializa la matriz de salida con «espacios gráficos». La línea 1050 determina si ha llegado el momento de poner un valor en el eje vertical. Si es así, las líneas 1060 a 1080 calculan su valor, eliminan los ceros residuales y justifican a la derecha la cantidad.

Las líneas 1090 a 1130 forman el bucle funcional. Por cada número de columna (CN) se obtiene el valor de FN Y y se determina si este valor corresponde a la línea en impresión. Si es así se introduce un valor en A%. Las líneas 1150 y 1160 añaden un RETURN y terminan la línea enviando A% a la impresora.

Cuando todas las líneas (50 en total, es decir, 350 puntos) han sido impresas, las líneas 1170-1200 dibujan el eje horizontal. Imprimir los números en este eje es algo más laborioso que en el vertical porque se trata de centrar los números en vez de justificarlos a la derecha. Las líneas 1210 a 1250 realizan esta función.

Finalmente las líneas 1260 a 1280 centran el título y lo imprimen en doble ancho.

MEJORA DE LA RUTINA

El defecto más grave de esta rutina es su lentitud (el dibujo de la figura 5 tarda unos 20 minutos). Si se dispone de un VIC con expansión se puede usar como programa principal el de la figura 6, cambiando la línea 1100 de la rutina por la siguiente:

```
1100 RS=RV—VA(CN):
IFRS<0THEN1130
```

Con estas modificaciones se ha realizado el dibujo de la figura 7. Aunque la mejora en cuanto a tiempo no es muy dramática. ■

CLUBS DE USUARIOS

trabajando con ampliaciones de memoria

por **JAUME JULIÁ**
(del Club de Usuarios de Ordenadores Commodore de Barcelona)

Como nuestros lectores deben haber observado, los «CLUBS DE USUARIOS» —que habían caído en un ominoso silencio— empiezan a moverse de nuevo. Esperamos que cunda el ejemplo de forma masiva y que dentro de poco no nos quepan en nuestra Revista los originales y las noticias de los que están en la mejor situación para "trabajar" los secretos de nuestros ORDENADORES PERSONALES. Como el presente ejemplo —que no dudamos en publicar a pesar de tocar este mismo tema en este número, debido a su calidad— que esperamos resuelva no pocos problemas a nuestros sufridos usuarios.

Las ampliaciones de memoria de 8 y 16K modifican una serie de zonas en la memoria del ordenador. La memoria de pantalla pasa de las posiciones 7680-8191 a las 4096-4607, el inicio de la memoria de color pasa de la posición 38400 a la 37888 y, por último, el inicio de la zona destinada a programas en BASIC pasa de la posición 4096 a la 4608. Todo este «desbarajuste» implica que la mayoría de programas pensados para trabajar con la configuración mínima no funcionen correctamente al conectar una expansión de memoria. Por otro lado, si deseamos utilizar un generador de caracteres propio con 8 ó 16K en el VIC, no nos sirve ninguna de las formas explicadas hasta ahora en la Revista.

Con el presente artículo pretendo solventar los problemas antes mencionados. Entra en tu VIC el programa número 1 y guárdalo (SAVE) en tu cassette o unidad de disco **ANTES** de ejecutarlo. Una vez hayas hecho esto, ponlo en marcha (RUN). En la pantalla aparecerán dos opciones:

—Pulsando la tecla de función F1, el ordenador quedará configurado con la memoria, posición de pantalla y color e inicio de la zona de programa que tendría sin ninguna expansión conectada, con lo que funcionarán todos los programas pensados para trabajar con 3.5K de memoria.

—Pulsando la tecla de función F7, se reserva una zona de memoria (4096-7679) para el generador de caracteres, subrutinas en lenguaje máquina, pantallas alternativas, etc. Esta zona está completamente protegida contra la «invasión» de los programas en BASIC. La pantalla se sitúa en las posiciones 7680-8191 y la memoria de color a partir de la posición 38400. Estas posiciones son las mismas que tiene el VIC sin ninguna expansión conectada, por lo que estarás acostumbrado a manejarlas y te será más cómodo programar.

En el caso de que desees crear un generador de caracteres en RAM, debes situarlo a partir de la posición

PROGRAMA NÚMERO 1

```
READY.

10 POKE36866,150:POKE648,30

20 FORJ=217TO228:POKEJ,158:NEXT

30 FORJ=229TO250:POKEJ,159:NEXT

40 GOTO130

50 POKE56,30:CLR:POKE4096,0:POKE44,
16:PRINT"[ CLR ]**CONFIGURACION BAS
E**"

60 PRINT"3583 BYTES LIBRES":NEW

70 POKE8192,0:POKE44,32:PRINT"[ CLR ]
COLOQUE EL GENERADOR APARTIR DE 409
6"

80 PRINT"[CRSRD]ACTIVELO EN SU PROG
RA-MA MEDIANTE:"

90 IFFRE(0)>14000THENBY=16384:GOTO1
10

100 BY=8192

110 PRINT"[CRSRD][ RED ]POKE 36869,
252"

120 PRINT"[CRSRD][CRSRD]DISPONE DE"
BY"BYTES":PRINT"PARA SU PROGRAMA.":
NEW

130 POKE36869,242

140 PRINT"[ CLR ][SHIFR][SHIFE][SHIFA]
[SHIFJ][SHIFU][SHIFS][SHIFT][SHIFE]
 [SHIFD][SHIFE] [SHIFM][SHIFE][SHIFM]
[SHIFO][SHIFR][SHIFI][SHIFA]"

150 IFFRE(8)<8000THENPRINT"[CRSRD][CRSRD]
[SHIFE]STE PROGRAMA NO PUEDEFUNCION
AR CON LA MEMO-RIA ACTUAL":END

160 PRINT"[CRSRD][CRSRD][SHIFS]I DE
SEA TENER LA CON-FIGURACION DE MEMO
RIA BASE PULSE '[SHIFF]1'."

170 PRINT"[CRSRD][SHIFS]I DESEA TRA
BAJAR CON EL GENERADOR DE CARAC-TER
ES A PARTIR DE LA  ";

180 PRINT"POSICION 4096 PULSE   '[SHIFF]
7'."

190 GETA$:IFA$<>"[  F1 ]"ANDA$<>"[  F7 ]
"THEN190

210 IFA$="[  F1 ]"THENPOKE36869,240
:GOTO50

220 IFA$="[  F7 ]"THENPOKE36869,240
:GOTO70

READY.
```

4096. (El funcionamiento y construcción de un generador de caracteres se ha explicado exhaustivamente en otros números, por lo que no entro en detalles sobre este tema). Una vez tengas listo el generador — colocados todos los valores que lo forman a partir de la posición 4096 — lo único que tendrás que hacer en tu programa, para activarlo, es colocar la instrucción **POKE 36869,252.** Para volver al generador normal en ROM usa la siguiente instrucción: **POKE 36869,240.** Olvídate de las posiciones 55 y 56, utilizadas en la creación de generadores con la memoria standard o la expansión de 3K. Para crear un generador con expansiones de 8 ó 16K no debes utilizar estas posiciones para nada.

Por último, el programa número 2 es una versión «comprimida» de una parte del programa número 1. Se encarga de colocar al VIC en la configuración de memoria standard y después cargar y ejecutar el programa que quieras. Mi consejo es que coloques este pequeño programa **ANTES** de cada uno de los programas que tengas que funcionen con 3.5K. De esta forma, cada programa constará de dos: el escrito aquí y el propio programa, o programa principal. De esta forma, todos los programas que funcionen con 3.5K funcionarán con **CUALQUIER** expansión de memoria, sin necesidad de desconectarla. Espero que este artículo te haya servido para comprender un poco más el diseño del ordenador y te sea útil para evitar los problemas que conllevan las expansiones de 8 y 16K.

PROGRAMA NÚMERO 2

```
READY.

10 POKE36869,240:POKE36866,150:POKE
648,30

20 FORJ=217TO228:POKEJ,158:NEXT

30 FORJ=229TO250:POKEJ,159:NEXT

50 POKE56,30:CLR:POKE4096,0:POKE44,
16:POKE198,2:POKE631,147:POKE632,13
1:NEW

READY.
```

A continuación voy a comentar brevemente la función de las líneas clave del programa número 1, ya que en ellas hay cosas muy interesante que te pueden ser de utilidad en otros programas.

LÍNEA (S)	DESCRIPCIÓN
10-30	Estas líneas se encargan de trasladar la posición de memoria de pantalla de las 4096-4607 a las 7680-8191. Con esta técnica se pueden usar varias pantallas en un programa, para animación, gráficos, etc.
50-60	Colocan la configuración mínima en el VIC. POKE56,30 limita por arriba la memoria, de forma que el VIC-20 no pueda acceder en BASIC a posiciones mayores de 7679. POKE 44,16 indica al ordenador que la zona destinada al BASIC se inicia en la posición 4096 (256*16=4096). Como que el primer byte de un programa debe contener el valor cero hacemos POKE 4096,0. Después de los PRINT, en la línea 60 se encuentra la instrucción NEW, necesaria siempre que movamos el inicio de la zona destinada al BASIC.
70-120	Reserva una zona de memoria desde 4096 a 7679. Esto lo realiza la instrucción POKE 44,32 que indica al VIC el principio de la zona destinada al BASIC. En este caso dicha zona empieza en la posición 8192 (32*256=8192) al igual que antes debemos colocar el valor cero en el primer byte de la zona de programa, por lo que hacemos POKE 8192,0. Después se comprueba mediante la memoria disponible si hay 8 ó 16K conectados, para imprimir el correspondiente mensaje.
130	Pasamos a modo minúsculas.
150	Comprueba la memoria conectada. Si es menor de 8K, el programa finaliza, ya que no puede funcionar.
190-220	Comprueba si hemos pulsado F1 ó F7 para pasar — después de volver al modo mayúsculas — a la parte del programa encargada de modificar la memoria del VIC.

NOTA: Para comprobar la tecla pulsada en el programa se han escrito comillas y la tecla de función correspondiente, lo que da un símbolo invertido. Si tienes el cartucho de ayuda al programador conectado, este sistema no sirve. Sustituye las teclas de función por otras (por ejemplo, los números 1 y 2). ■

TABLAS DE CÓDIGO ASCII y PET

por **JORDI SASTRE**

La TABLA 1 contiene los códigos «ASCII» de Commodore. La primera columna corresponde al valor decimal, la segunda al hexadecimal, y la tercera es el carácter ASCII correspondiente a cada código. A veces aparecen dos caracteres equivalentes a un solo valor ASCII; el primero corresponde a la visualización en pantalla estando el ordenador en caja baja (minúsculas/mayúsculas) y el segundo en caja alta (mayúsculas/gráficos).

Ejemplo: PRINT «$» y PRINT CHR$(36) imprimen el signo de dólar.

PRINT CHR$(65) imprime «a» o «A» según la pantalla esté en caja baja o alta, respectivamente.

PRINT CHR$(147) borra la pantalla.

* — Estos caracteres de control sólo están disponibles en los modelos CBM-8032 y CBM-4032 en versión de pantalla de 12 pulgadas.

2 — Nótese que los caracteres 96 a 127 y 224 a 255 son repeticiones de los 32 a 63 y 160 a 191. Así, PRINT CHR$(98) imprime el carácter de comillas ("), pero no activa su flag especial de comillas.

3 — Caracteres sólo válidos en el CBM-8032.

La TABLA 2 es idéntica a la Tabla 1, salvo que en este caso se trata de los códigos de la pantalla, o código PET. El código ASCII es el que se utiliza cuando se imprime mediante PRINT, el código PET corresponde a la impresión por POKE (o consulta por PEEK). Por ejemplo, para escribir en la pantalla una «A», se puede usar PRINT CHR$(193) o POKE x,65, siendo x el byte de pantalla deseado para la visualización.

La TABLA 3 es una rápida referencia de los gráficos CBM indicando la tecla a la que corresponden (no válida para VIC ni C-64), el Código ASCII (ver Tabla 1) y el código PET (ver Tabla 2).

MARKETCLUB

- Programas VIC-20 y COMMODORE 64. Paquetes standard de clientes, artículos, etiquetas, gestión de cobros/pagos, etc... Entrega inmediata. Programas a medida, plazo máximo de entrega: 10 días. Tel. 345 10 00, Srta. M. José (mañanas) o 345 87 75 Sr. Martínez (fuera de horas de oficina o sábados mañana).
- Vendo VIC-20 con unidad de cassette y los dos cursos de BASIC, todo por 35.000 ptas. Interesados llamar a RAFAEL. Tel.: (93) 332 08 95.
- Vendo módulo para VIC/20 compuesto de: dos puertas de ocho entradas/salidas más dos de control y estado. Dos temporizadores programables. Posibilidad de interrupción IRQ-NMI. Con las correspondientes instrucciones. Razón: Luis Torrents, c/ Velázquez, 39 - TERRASSA (Barcelona).
- Hago programas en BASIC COMMODORE bajo encargo. Vendo VIC-20, con 16 K de RAM, cartuchos de: Ayuda al Programador, Monitor Lenguaje Máquina, Superexpander + 3 K y tres juegos. Además se darán al comprador muchos programas hechos y comprados y 5 libros existentes o no en España sobre el VIC. Todo por 65.000 ptas. (vale más de 100.000 ptas.) Por separado también. Llamar al (91) 253 13 40. Horas comida y cena. Dirección: Francisco Gutiérrez. Santiago Rusiñol, 12 - MADRID-3.
- Compraría equipo COMMODORE CBM serie 3000 (CPU-3032 y FLOPPY-3040), usado y en buenas condiciones. Llamar por la noche al (951) 23 86 72 o escribir al apartado número 286 de Almería.

EL CLUB DE USUARIOS DE ORDENADORES COMMODORE EMPIEZA A MOVERSE

Desde su nacimiento — hace aproximadamente cuatro meses — se han lanzado muchos proyectos, algunos de los cuales son ya felizmente realidad. A continuación enumeramos los principales:

— Biblioteca y programoteca.
— Suscripción a revistas extranjeras.
— Formación de equipos de trabajo para profundizar en aspectos específicos del ordenador y desarrollo de programas.
— Intercambio de ideas y experiencias entre los socios.
— Organización de cursillos: iniciación al VIC-20 y C-64, el BASIC de Commodore, introducción al lenguaje máquina, etc.

Entre los servicios a los socios destacamos:

— Acceso a la biblioteca. (Los socios que no vivan en Barcelona podrán recibir fotocopias de los artículos de su interés.)
— Intercambio de programas.
— Recepción periódica de circulares informativas con programas, trucos, POKES, etcétera.
— Mercado de ocasión: Compra, Venta, Cambio...
— Descuentos especiales en los cursillos, accesorios y software.
— Resolución por parte de nuestro equipo técnico de dudas sobre el ordenador y problemas de programación.

Además se estudiarán todas las sugerencias para lograr un club «a la medida» de la mayoría de los socios.

En la actualidad nos reunimos semanalmente en una tienda de informática, que nos ha cedido desinteresadamente sus instalaciones. Sin embargo, estamos gestionando el alquiler de un local para disponer de más espacio y comodidad, con lo que podremos ampliar todavía más nuestros objetivos.

El importe de la inscripción anual es de 3.000 pesetas. Los socios que no vivan en Barcelona obtendrán, en compensación por no poder asistir a las reuniones, vales por 200 fotocopias gratuitas del material de nuestra biblioteca. Además, al formalizar la inscripción, los nuevos socios recibirán una cassette conteniendo programas muy útiles. (Tenemos lista la cassette para el VIC-20 y a punto de terminar la del Commodore 64.)

NOTA: Las actividades del Club están centradas principalmente en los ordenadores VIC-20 y C-64, aunque se admiten usuarios de CUALQUIER ordenador «Commodore».

Para formalizar la inscripción, rellena el cupón adjunto y envíalo junto con un talón o giro postal por el importe de la inscripción ANUAL, a CLUB DE USUARIOS DE ORDENADORES COMMODORE. - Vía Augusta, 120 - Barcelona-6. (Ésta es la dirección PROVISIONAL del Club. La definitiva será comunicada cuando hayamos encontrado el local apropiado.)

Un último consejo: Aunque no vivas en Barcelona, si en tu lugar de residencia no dispones de Club de Usuarios, inscríbete en éste. Además de las ventajas ya enumeradas, el Club se encargará de poner en contacto a todos los socios de cada población, con lo que más adelante podréis formar vuestro propio Club.

Te esperamos.

CLUB DE USUARIOS DE ORDENADORES "COMMODORE"

D.:

Dirección:

Población: D.P. Provincia: Tel:

Se afilía al Club de Usuarios de Ordenadores «Commodore» por un año prorrogable, mediante el pago de 3.000 PESETAS, importe de la cuota anual. Esta cuota me da derecho a disfrutar de los servicios y actividades que organice el Club.

FIRMA DEL PADRE (Menores de 18 años) FIRMA

Ordenador que poseo:

TABLA 2

0	00	@	32	20	b̸	64	40		96	60	128	80	160	A0	192	C0	224	E0
1	01	a A	33	21	!	65	41	A	97	61	129	81	161	A1	193	C1	225	E1
2	02	b B	34	22	"	66	42	B	98	62	130	82	162	A2	194	C2	226	E2
3	03	c C	35	23	#	67	43	C	99	63	131	83	163	A3	195	C3	227	E3
4	04	d D	36	24	$	68	44	D	100	64	132	84	164	A4	196	C4	228	E4
5	05	e E	37	25	%	69	45	E	101	65	133	85	165	A5	197	C5	229	E5
6	06	f F	38	26	&	70	46	F	102	66	134	86	166	A6	198	C6	230	E6
7	07	g G	39	27	'	71	47	G	103	67	135	87	167	A7	199	C7	231	E7
8	08	h H	40	28	(	72	48	H	104	68	136	88	168	A8	200	C8	232	E8
9	09	i I	41	29	)	73	49	I	105	69	137	89	169	A9	201	C9	233	E9
10	0A	j J	42	2A	*	74	4A	J	106	6A	138	8A	170	AA	202	CA	234	EA
11	0B	k K	43	2B	+	75	4B	K	107	6B	139	8B	171	AB	203	CB	235	EB
12	0C	l L	44	2C	,	76	4C	L	108	6C	140	8C	172	AC	204	CC	236	EC
13	0D	m M	45	2D	–	77	4D	M	109	6D	141	8D	173	AD	205	CD	237	ED
14	0E	n N	46	2E	.	78	4E	N	110	6E	142	8E	174	AE	206	CE	238	EE
15	0F	o O	47	2F	/	79	4F	O	111	6F	143	8F	175	AF	207	CF	239	EF
16	10	p P	48	30	0	80	50	P	112	70	144	90	176	B0	208	D0	240	F0
17	11	q Q	49	31	1	81	51	Q	113	71	145	91	177	B1	209	D1	241	F1
18	12	r R	50	32	2	82	52	R	114	72	146	92	178	B2	210	D2	242	F2
19	13	s S	51	33	3	83	53	S	115	73	147	93	179	B3	211	D3	243	F3
20	14	t T	52	34	4	84	54	T	116	74	148	94	180	B4	212	D4	244	F4
21	15	u U	53	35	5	85	55	U	117	75	149	95	181	B5	213	D5	245	F5
22	16	v V	54	36	6	86	56	V	118	76	150	96	182	B6	214	D6	246	F6
23	17	w W	55	37	7	87	57	W	119	77	151	97	183	B7	215	D7	247	F7
24	18	x X	56	38	8	88	58	X	120	78	152	98	184	B8	216	D8	248	F8
25	19	y Y	57	39	9	89	59	Y	121	79	153	99	185	B9	217	D9	249	F9
26	1A	z Z	58	3A	:	90	5A	Z	122	7A	154	9A	186	BA	218	DA	250	FA
27	1B	[	59	3B	;	91	5B		123	7B	155	9B	187	BB	219	DB	251	FB
28	1C	\	60	3C	<	92	5C		124	7C	156	9C	188	BC	220	DC	252	FC
29	1D	]	61	3D	=	93	5D		125	7D	157	9D	189	BD	221	DD	253	FD
30	1E	↑	62	3E	>	94	5E		126	7E	158	9E	190	BE	222	DE	254	FE
31	1F	←	63	3F	?	95	5F		127	7F	159	9F	191	BF	223	DF	255	FF

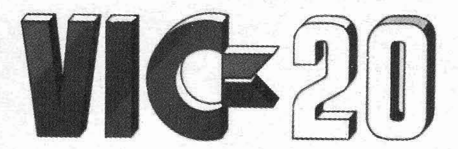
shifted (64–127) reversed (128–255) shifted (192–255)

TABLA 3

CROSS-REFERENCE TO CBM GRAPHICS CHARACTERS

KEY:	sh-$	sh-r	sh-f	sh-@	sh-c	sh-d	sh-e	sh-#
CHR$:	228	210	198	192	195	196	197	227
POKE:	100	82	70	64	67	68	69	99
KEY:	sh-%	sh-t	sh-g	sh-b	sh-]	sh-h	sh-y	sh-'
CHR$:	229	212	199	194	221	200	217	231
POKE:	101	84	71	66	93	72	89	103
KEY:					REVERSE, CHR$(18), THEN-			
KEY:	sh-$	sh-/	sh-9	sh-"	sh-8	sh-7	sh-#	sh-space
CHR$:	228	239	249	226	184	183	163	160
POKE:	100	111	121	98	248	247	227	224
KEY:					REVERSE, CHR$(18), THEN-			
KEY:	sh-%	sh-4	sh-5	sh-!	sh-6	sh-*	sh-'	sh-space
CHR$:	229	244	245	225	182	170	167	160
POKE:	101	116	117	97	246	234	231	224
KEY:	sh-o	sh-p	sh-:	sh-l	sh-v	sh-[	sh-m	sh-n
CHR$:	207	208	186	204	214	219	205	206
POKE:	79	80	58	76	86	91	77	78
KEY:	sh-=	sh--	sh-Ø	sh-.	sh-1	sh-2	sh-3	sh-+
CHR$:	189	173	176	174	177	178	179	171
POKE:	125	109	112	110	113	114	115	107
KEY:	sh-<	sh->	sh-,	sh-;	sh-?			
CHR$:	190	188	172	187	191			
POKE:	126	124	108	123	127			
KEY:	sh-k	sh-j	sh-u	sh-i	sh-w	sh-q		
CHR$:	203	202	213	201	215	209		
POKE:	75	74	85	73	87	82		
KEY:	sh-)	sh-←	sh-&	sh-(	sh-\			
CHR$:	169	223	166	168	220			
POKE:	105	95	102	104	92			
KEY:	sh-a	sh-s	sh-z	sh-x	sh-↑			
CHR$:	193	211	218	216	222			
POKE:	65	83	90	88	94			

Depósito Legal Sep. B. 2133-1958 - Altés, s.l., Caballero 87, Barcelona



Microprocesador: 6502 de MOS TECHNOLOGY de 8 bits.
Memoria: 5 Kbytes de RAM ampliables a 32 K 20 Kbytes de ROM ampliables a 28 K
Pantalla: 23 lineas de 22 caracteres Modulador para conectar a un televisor normal. Salida para monitor de video.
Colores: 8 para el marco, 16 para el fondo de la pantalla y ocho para los caracteres individuales, video inverso.
Gráficos: Semi-gráficos por teclado y alta resolución por redefinición del generador de caracteres (situándolo en RAM). Definición de 176 por 184 puntos.
Teclado: Tipo QWERTY de 62 teclas más cuatro de función definibles por el usuario.
Sonido: Tres voces de tres octavas cada una decaladas una octava entre sí, resultando una extensión total de cinco octavas. Un generador de ruido aleatorio afinable para efectos especiales, un control general de volumen.
Programación: Lenguaje BASIC, intérprete residente en ROM de 8 K. Posibilidad de interceptar las funciones del Basic para crear nuevas instrucciones «a medida». El Basic del Vic es uno de los más rápidos actualmente en el mercado.
Complementos: Port de usuario de 8 bits entrada/salida más dos señales de sincronismo.
Bus de expansión para ampliaciones de memoria y periféricos.
Port de juegos con conexión para dos potenciómetros (paddles), y una palanca de juegos (joystick).
Almacenamiento de masa: Unidad de cassette C2N de diseño especial para registrar programas y datos (ficheros secuenciales).

VIC-1540 UNIDAD DE DISCO

Capacidad total: 174848 bytes por disco.
Secuencial: 168656 bytes por disco.
Entradas de directorio: 144 por disco.
Sectores por pista: De 17 a 21.
Bytes por sector: 256.
Pistas: 35.
Bloques: 683 (644 bloques libres).
Soportes de información: Discos estandar de 5 1/4 pulgadas, de una sola cara y densidad simple.
Sistema operativo: DOS de COMMODORE inteligente (tiene procesador propio y no ocupa memoria del ordenador central).

VIC-1515 IMPRESORA

Método de impresión: Matriz de 5×7 puntos, impacto por un solo martillo.
Modo caracteres: Mayúsculas y minúsculas, símbolos, números y caracteres gráficos del VIC-20.
Modo gráfico: Puntos direccionables (bit image). Siete puntos verticales por columna, 480 columna máximo.
Velocidad: 30 caracteres/segundo, de izquierda a derecha, unidireccional.
Caracteres/Linea: Máximo 80. (Posibilidad de impresión en doble ancho).
Espaciado entre lineas: 6 lineas/pulgada -modo caracteres, 9 lineas/pulgadas - modo gráfico.
Velocidad de salto de lineas: 5 saltos/seg. -modo caracteres, 7,5 saltos/seg. - modo gráfico.
Alimentación de papel: Arrastre por tractor.
Ancho de papel: Entre 4,5 y 8 pulgadas.
Copias: Original más dos copias.

CARTUCHOS

Ayuda programador: Este cartucho facilita la edición y depuración de programas en Basic. Instrucciones y comandos: RENUMBER, MERGE, FIND, CHANGE, DELETE, AUTO, TRACE, STEP, OFF, KEY, EDIT, PROG, DUMP, HELP y KILL.

Super expander: Intercepta el Basic del VIC permitiendo incrementar sus instrucciones y comandos en aplicaciones gráficas, de sonido y juegos. Instrucciones y comandos: KEY, GRAPHIC, COLOR, POINT, REGION, DRAW, CIRCLE, PAINT, CHAR, SCNCLR, SOUND, RGR, RCOLR, RDOT, RPOT, RPEN, RJOY y RSND.

Monitor de lenguaje máquina: Este monitor altamente sofisticado facilita enormemente la depuración de programas en lenguaje máquina, es ideal como complemento del Basic para redactar y poner en marcha rutinas de alta velocidad y manejo de datos en tiempo real. Instrucciones y comandos: ASSEMBLE, BREAKPOINT, DISASSEMBLE, ENABLE VIRTUAL ZERO PAGE, FILL MEMORY, GO, HUNT, INTERPRET, JUMP TO SUBROUTINE, LOAD, MEMORY, NUMBER, QUICK TRACE, REGISTERS, REMOVE BREACPOINTS, SAVE, TRANSFER, WALK y EXIT TO BASIC.

Además existen cartuchos de ampliación de memoria de 3, 8 y 16 Kbytes.

CURSO DE INTRODUCCION AL BASIC PARTE I:

En forma de libro se ha editado la primera parte de un curso de Basic que parte «de cero» y está basado en el VIC-20. Va acompañado de dos cassettes con programas y ejercicios para autocontrol de los progresos en el aprendizaje.

MODULO DE EXPANSION DE MEMORIA:

Acabado en metal de gran robustez, permite la conexión de un máximo de 6 cartuchos simultáneamente, aloja al VIC y al modulador de video y permite colocar encima el televisor, tiene alojamiento para accesorios y asegura una óptima conexión del VIC a sus periféricos.







microelectrónica
y control, s.a.

Taquigrafo Serra, 7 5.° Telf. 250 51 03. BARCELONA-29
Princesa, 47 3.° G. Telf. 248 95 70. MADRID-8